**Helm Architecture Overview**

1. **Helm Client**:
   * The command-line interface (CLI) tool used to manage Helm charts and releases.
   * Executes commands like install, upgrade, delete, etc.
   * Communicates with the Kubernetes API server.
2. **Helm Charts**:
   * Collections of Kubernetes YAML templates, organized as a package.
   * Contain default configurations (values.yaml), templates, and metadata (Chart.yaml).
3. **Kubernetes Cluster**:
   * Helm interacts with the Kubernetes API server to deploy and manage resources in the cluster.
4. **Helm Repository**:
   * A collection of Helm charts stored remotely or locally.
   * helm repo commands manage these repositories.

**1. Chart Management**

Commands for managing chart repositories and chart files.

* **helm repo add**: Adds a chart repository URL to the Helm client.
  + Component: Helm Client → Repository.
* **helm repo update**: Fetches updated chart metadata from repositories.
  + Component: Helm Client → Repository.
* **helm search repo**: Searches locally cached metadata for charts in repositories.
  + Component: Helm Client → Repository Metadata.

**2. Deployment Management**

Commands for installing, upgrading, or deleting Helm releases.

* **helm install**: Deploys a chart to the cluster by rendering templates and creating Kubernetes resources.
  + Component: Helm Client → Kubernetes API Server → Cluster Resources.
* **helm upgrade**: Updates an existing release by applying changes to templates or configurations. Helm will tacke are of all resources with in the chat and update like IMAGE, cluster components versions. databases will remain as same as we will right in chat to ignore it.
  + Component: Helm Client → Kubernetes API Server → Cluster Resources.
* **helm uninstall**: Deletes a Helm release and removes related resources from the cluster.
  + Component: Helm Client → Kubernetes API Server → Cluster Resources.

**3. Release Management**

Commands for inspecting, modifying, or debugging deployed releases.

* **helm list**: Lists all Helm releases in a namespace or cluster-wide.
  + Component: Helm Client → Kubernetes API Server → Helm Release Metadata.
* **helm status**: Displays the current status of a release.
  + Component: Helm Client → Kubernetes API Server → Cluster Resources.
* **helm history**: Shows the revision history of a release.
  + Component: Helm Client → Kubernetes API Server → Helm Release Metadata.

**4. Debugging and Validation**

Commands to preview and validate configurations.

* **helm template**: Renders chart templates locally without deploying to the cluster.
  + Component: Helm Client → Local Chart Files.
* **helm lint**: Validates the structure and syntax of a chart.
  + Component: Helm Client → Local Chart Files.
* **helm install --dry-run**: Simulates an install, rendering templates and displaying planned Kubernetes resources without applying them.
  + Component: Helm Client → Local Chart Files.

**5. Chart Creation and Packaging**

Commands for creating or packaging charts.

* **helm create**: Generates a scaffold for a new chart.
  + Component: Helm Client → Local File System.
* **helm package**: Bundles a chart directory into a .tgz package.
  + Component: Helm Client → Local File System.
* **helm show values**: Displays the default values of a chart.
  + Component: Helm Client → Chart Metadata

**Repository Management**

1. helm repo add <repo\_name> <repo\_url> - Add a Helm repository.
2. helm repo update - Update all Helm repositories.
3. helm repo list - List all added repositories.
4. helm repo remove <repo\_name> - Remove a Helm repository.
5. helm search repo <chart\_name> - Search for charts in repositories.
6. helm search hub <chart\_name> - Search for charts in Helm Hub.

**Chart Deployment**

1. helm install <release\_name> <chart\_name> - Install a chart as a release.
2. helm install <release\_name> <chart\_name> --set key=value - Install with custom settings.
3. helm install <release\_name> <chart\_name> -f <values\_file.yaml> - Install with a custom values file.
4. helm upgrade <release\_name> <chart\_name> - Upgrade an existing release.
5. helm uninstall <release\_name> - Uninstall a release.

**Release Management**

1. helm list - List all installed releases.
2. helm status <release\_name> - Show the status of a release.
3. helm rollback <release\_name> <revision> - Roll back a release to a specific revision.
4. helm history <release\_name> - Show the revision history of a release.

**Debugging and Validation**

1. helm lint <chart\_directory> - Validate the structure of a chart.
2. helm template <release\_name> <chart\_name> - Render templates without installing.
3. helm install <release\_name> <chart\_name> --dry-run - Simulate an install without deploying.
4. helm upgrade <release\_name> <chart\_name> --dry-run - Simulate an upgrade.

**Chart Management**

1. helm show chart <chart\_name> - Show details about a chart.
2. helm show values <chart\_name> - Display the default values of a chart.
3. helm show all <chart\_name> - Show all details about a chart.
4. helm pull <chart\_name> - Download a chart locally.
5. helm package <chart\_directory> - Package a chart into a .tgz file.

**Chart Creation**

1. helm create <chart\_name> - Scaffold a new Helm chart.

**Helm Help and Version**

1. helm version - Display Helm's version.
2. helm help - Display help for Helm commands.